ISD Software Requirements Development

Number: 580-PC-020-001
Effective Date: June 20, 2005
Expiration Date: June 20, 2010

Approved By: (signature)
Name: Joe Hennessy
Title: Chief, ISD

Responsible Office: 580/Information Systems Division (ISD)
Title: Software Requirements Development

Asset Type: Process
PAL Number: 2.2.1

Purpose

The purpose of software requirements development process is to establish a documented common understanding of customer needs, intended product use, needed software capabilities, development resources and constraints.

This process describes the tasks necessary for the collection, definition, and documentation of detailed software requirements.

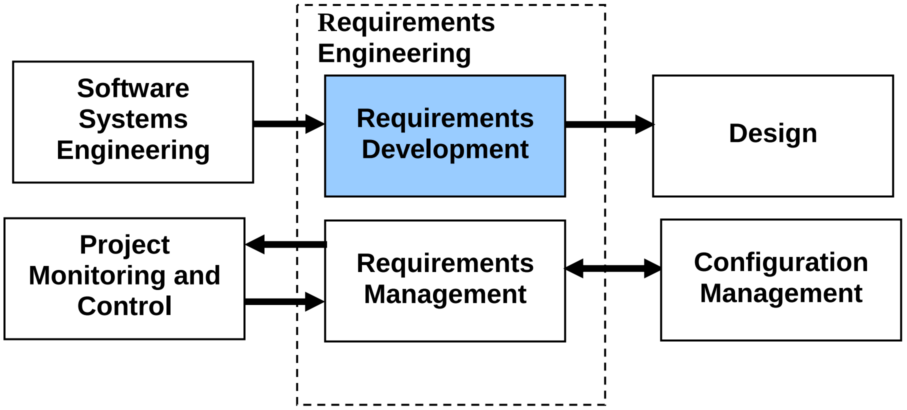
The result of this process is an organized set of requirements for a software product development effort that:

- Supports the customer's needs, goals, and objectives
- Remains within a well-defined scope
- Supports tracking and recording of costs and efforts
- Identifies the necessary resources and constraints
- Develops a software requirements database

Scope

This process applies to the development of ISD mission software, typically of classes A, B, and C as defined in NPR 7150.2 located at http://nodis.hq.nasa.gov/.

Context Diagram



Roles and Responsibilities

Customer

- Provides needs, expectations, constraints, and resources
- Provides high-level requirements
- May define external interfaces
- Evaluates requirements developed.

Author

- Collects and researches high-level requirements from the customer
- Translates customer needs and high-level requirements into detailed customer and product requirements
- Documents refined operational concepts
- Documents requirements and specifications
- Documents interfaces

GUIDANCE: The author is often the Product Development Lead (PDL) or a PDL designee.

Development Engineer (a.k.a. Software Developer)

- Provides the Author with insight into approaches and feasibility.
- Evaluates Commercial-off-the-shelf (COTS) and Government-off-the-shelf (GOTS) products and proposes re-use candidates
- Identifies interfaces

GUIDANCE Often more than one software developer is used for various aspects of the work. On small projects the role may be filled by the development team lead or by consultation with senior staff.

Interface Representative

- Provides a point of contact authorized to commit to effort from an external entity that will supply or support an identified interface.

GUIDANCE: This role may be held by a manager or a technical expert.

Reviewer

- Validates requirements as to consistency and feasibility
- Verifies requirements as to satisfying needs for the reviewer's area of knowledge and expertise.

GUIDANCE: This is a member of a team including the CCB, PDL, Customer, Software Assurance Engineer, and other stakeholders.

Entry Scenarios

- This process is entered at the initiation of a mission software development project.

Inputs

- High Level Requirements
- System Operations Concepts
- Customer Needs

GUIDANCE: High level requirements typically consist of:

- *System or subsystem requirements document*
- *Software requirements document*
- *Interface Requirements/Control Documents (IRDs / ICDs)*
- *Assurance Requirements*

High level requirements may be supplied to the project externally or developed through project activities. Requirements documents that are input to project planning may be in draft form. These requirements explicitly or *implicitly limit the scope of the software project, state needs and goals and explicitly identify some interfacing elements. All documents used need to be noted.*

If requirements documents are unavailable, requirements definition can begin when the existence of the customer's or user's needs have been documented. A statement of customer needs is usually in the form of a Memorandum of Agreement (or Understanding), a contract, or Statement of Work.

Operational Concepts are a set of working definitions, situations and functions that are expected to arise or potentially arise in use of the proposed product. Operational scenarios may be developed jointly after requirement definition is initiated and are a proven tool for capturing requirements, needs, work and data flow, and use cases using ordinary language and diagrams.

Customer needs are usually evoked through interviews and discussions of the Operations Concepts.

Entry Criteria

- High-level requirements are available

OR

- Statement of customer needs is available

Exit Criteria

- The documented requirements are validated, approved and baselined.

Outputs

- Baselined, validated requirements

GUIDANCE: Should include the following:

- *Baseline documents*
- *Approval by affected parties*
- *Allocation of responsibility for requirements to software, people, hardware, and subsystems*

- Interface documents

GUIDANCE: These documents should:

- *Identify all external interfaces*
- *Provide a contact person for each interface*
- *Describe items to be transferred across the interface*
- *Describe test time and equipment needed*
- *Provide a schedule of tests*

- Reuse plans

GUIDANCE: Specify candidate items for re-use, associated costs and benefits.

- Traceability Matrices

GUIDANCE: Provide bi-directional tracing for all requirements

- *Trace detailed requirements to all higher level source documents or documented customer needs.*

- *Trace detailed requirements down to validation method, verification method, and implementation allocation.*

- Operational Scenarios

GUIDANCE: Each scenario provides a description of how the software, hardware, users and interfaces will interact in a given circumstance. The discussion of both nominal and off-nominal circumstances allows the user and requirements gathering team to speak the same language and agree on observable behavior of the system.

- Historical Records

GUIDANCE: All actions resulting in minutes and notes are collected to be included in final project documentation. A history of changes and modifications to all documents should be kept.

Major Tasks

1. Analyze High Level Requirements [Author, Customer, Developer]
2. Define Detailed Requirements and Specifications [Author, Interface Representative, Developer]
3. Verify Requirements and Specifications [Author, Customer, Developer, Interface Representative, Reviewer]
4. Validate Requirements and Specifications [Author, Customer, Developer, Interface Representative, Reviewer, Approver]
5. Obtain Approval [Author, Approver]

Task 1:

Analyze High Level Requirements. [Author, Customer, Developer]

Guidance: The author leads the effort and works with the Customer and Development Engineer.

a) Examine, discuss, and understand the high level requirements, and Operations Concepts.

b) Identify the scope of the requirements, the purpose(s) of the software and analyze any constraints (e.g. budget, schedule or performance) affecting the software requirements.

c) Balance requirements scope to remain within programmatic constraints.

d) Develop and document operational scenarios.

e) Perform Reuse / COTS / GOTS study and document the results.

f) Recognize and document assumptions made in conducting the analysis.

GUIDANCE: Operational scenarios are recommended as a tool for involving the customer in discussion, comprehension, and analysis of high-level and detailed requirements. The emphasis of analysis is on functionality, external interfaces, performance, customer expectations, and design constraints. User interfaces particularly can benefit from prototyping during this phase to allow early refinement. The Importance of each high level function should be captured during analysis and interviews for use in strategizing development (i.e., essential, desirable, urgent.) The analysis itself may be iterative, especially when conflicts are discovered or when dealing with nebulous goals.

Hardware choices may be dictated as part of the high level requirements. Hardware / Software allocations and data flows are part of the analysis.

Task 2:

Define Detailed Requirements and Specifications. [Author, Interface Representative, Developer]

Guidance: The author coordinates with the user or his representative, interfacing system developers, and software developers to collect the necessary information to perform the following tasks producing: detailed requirements and specifications, a traceability matrix, and ICDs and IRDs as required. The user interface is included in defining interfaces:

a) Refine Operations Concepts and operational scenarios to insure all functionality is documented.

b) Expand the high-level requirements to detailed requirements. *GUIDANCE: The team identifies input, output, and processing needed to satisfy the high-level requirements using structured, object-oriented, or another analysis method focusing on what the software needs to do rather than specific "how-to" details.*

c) Define external interface requirements including any deviations from existing ICDs, IRDs, or Memos of Understanding (MOUs).

d) Allocate the detailed requirements and specifications to subsystems or major components.

e) Analyze detailed requirements to assure they are within programmatic constraints.

f) Trace detailed requirements bi-directionally to high level requirements and down to subsystems and possible verification methods in matrices.

GUIDANCE: Tracing down to actual verification methods and test sets should be done by the testing function at the same time as requirements development. Each detailed requirement also needs to be mapped down to a validation method, implementing subsystem, and back up to a source high level requirement. Tracing must be kept bi-directional by tracing each requirement to its source and maintaining all affected requirement source and destination documents.

g) Document the detailed requirements and specifications developed.

GUIDANCE: The purpose, scope, description and specific software requirements that will satisfy the high-level requirements and expressed customer needs are prepared as a document for approval.

Task 3:

Verify Requirements and Specifications. [Author, Customer, Developer, Interface Representative, Reviewer]

GUIDANCE: The Author coordinates a team including Customer, Developer, Interface Representative, Reviewer, and Approver to examine the available inputs and generated requirements. The team uses comprehensive methods to assure the generated requirements are consistent, complete, and feasible. Subtasks include the following steps which may be repeated as necessary and performed sequentially or in parallel.

a) Conduct requirements peer reviews to ensure agreement regarding the intent and purpose of each requirement and the reason for limits, tolerance and margin in each specification.

b) Clarify ambiguous requirements.

c) Determine the technical feasibility of each requirement and any risks inherent in candidate approaches.

GUIDANCE: Some items which may indicate risk include: incomplete *information, technology readiness level, schedule issues, cost issues, uncertainties, missing stakeholder input, and incompletely defined interfaces. Priority issues should be addressed during these discussions.*

d) Verify consistency, necessity, and completeness both internal to the requirements and against driving documents

GUIDANCE: Complete requirements should cover nominal and off-nominal interfacing software and hardware scenarios. Obtaining approval may require iterating tasks.

e) Model performance or prototype as needed.

Task 4:

Validate Requirements and Specifications. [Author, Customer, Developer, Interface Representative, Reviewer]

GUIDANCE: The Author coordinates a review for the benefit of all stakeholders including the Customer(s), Product Development Team, Interface Representative(s) and end users. The team uses comprehensive methods to assure the generated requirements are correct, complete, and feasible. Methods suggested include reviews, prototyping, hand calculation, and paper simulations.

a) Determine and document the method of validation to be used for each requirement.

b) Conduct a Software Requirements Review (SRR) to present the requirements to all stakeholders.

c) Collect and track issues resulting from the SRR as Review Item Dispositions (RIDs) or Requests for Action (RFAs).

d) Update the requirements as necessary to address RIDs or RFAs.

Task 5:

Obtain Approval. [Author, Approver]

a) Obtain signed approval to proceed with development.

GUIDANCE: Approvers should represent both customer and developers to assure concurrence.

b) Baseline the requirements and specifications.

c) Publish the Baselined requirements and specifications to all affected parties.

Measures

Recommended Measures:

d) Detailed Requirements Count.

a) Open and closed RIDs, and RFAs.

Tools and Templates

Name	Description
Contents of the Software Requirements Review	Checklist for Contents of the Software Requirements Review http://software/AssetsApproved/PA2.2.1.6.doc

Tools and Templates

Name	Description
CORE	Tool supporting model-based systems engineering and product design, including requirements analysis http://www.vitechcorp.com/CORE/productline.html
DOORs	Requirements tracing aid – http://www.telelogic.com/products/doorsers/doors/
FSW Requirements Document Template	http://software/AssetsApproved/PA2.2.1.2.1.doc
FSW Requirements Review Standard	http://software/AssetsApproved/PA2.2.1.6.1.doc
Rational Rose	Requirements tracing aid - http://www.rationalrose.com/
SLATE	Requirements tracing aid http://www.sdrc.com/

Training

Course Name	Description
Requirements Engineering Workshop	Available from Teraquest, 3-4 day course (ISD Training Plan) HQ001
Mastering the Requirements Process	Available from SQE, 3 day course (ISD Training Plan)
NPR 7150.2/IEEE 12207 Class	Dr. Lewis Gray describes a mapping of IEEE 12207 to NPR 7150.2.
Software Requirements Development	ISD Training Plan - ISD009
System Requirements	

References

- **ETVX Diagram:** A hyper-link to this diagram can be found in the Process Asset Library on-line version of this document.
- **Glossary:** http://software.gsfc.nasa.gov/glossary.cfm Defines common terms used in ISD processes
- **IEEE 830-1998: Recommended Practice for Software Requirements Specifications** (available through: http://standards.nasa.gov/npts/login.taf at http://standards.ieee.org/catalog/olis/se.html)
- **NPR: 7150.2: NASA Software Engineering Requirements** http://software.nasa.gov/npr_7150_2/index.cfm
- **Process Asset Library:** http://software.gsfc.nasa.gov/process.cfm Library of all ISD process descriptions
- **In-House Development And Maintenance Of Software Products – GPG 8700.5** http://gdms.gsfc.nasa.gov/gdmsnew/srv/GDMSNEWDatabaseObject?document_id=6152
- **SQ Software Specification Review (SSR) Product Checklist** http://sw-assurance.gsfc.nasa.gov/disciplines/quality/checklists/pdf/software_specification_review.pdf

- **SQ Software Requirements Specification (SRS) Document Checklist** http://sw-assurance.gsfc.nasa.gov/disciplines/quality/checklists/pdf/software_requirement_specification.pdf
- **Systems Engineering – GPG 7120.5A –** http://gdms.gsfc.nasa.gov/gdmsnew/srv/GDMSNEWDatabaseObject?document_id=6153

Quality Management System Records

Controlled Document / Description	Record Custodian
Requirements Document	Project Librarian
Traceability Matrix	Project Librarian

Change History

Version	Date	Description of Improvements
1.0	June 20, 2005	CCB Approved Baseline